



Autonomic Context-aware Management in Interplanetary Communication Systems

Peoples, C., Parr, G., Scotney, BW., & Adrian, M. (2011). Autonomic Context-aware Management in Interplanetary Communication Systems. *IEEE Aerospace and Electronic Systems Magazine*, 26(2), 26-33. <https://doi.org/10.1109/MAES.2011.5739487>

[Link to publication record in Ulster University Research Portal](#)

Published in:
IEEE Aerospace and Electronic Systems Magazine

Publication Status:
Published (in print/issue): 01/02/2011

DOI:
[10.1109/MAES.2011.5739487](https://doi.org/10.1109/MAES.2011.5739487)

Document Version
Publisher's PDF, also known as Version of record

General rights
Copyright for the publications made accessible via Ulster University's Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Ulster University's institutional repository that provides access to Ulster's research outputs. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact pure-support@ulster.ac.uk.

Autonomic Context-Aware Management in Interplanetary Communications Systems

C. Peoples, G. Parr, B. Scotney & A. Moore
University of Ulster

ABSTRACT

Maintaining connectivity in deep-space communications is of critical importance to key missions and the ability to adapt node behavior “on-the-fly” can have dynamic benefits. Autonomic operation minimizes failure risk by performing local configurations using collected context data and on-board policies, improving response time to events, and reducing remote mission management expense. Herein, we evaluate cost-benefit impacts when a context-aware brokering algorithm developed to achieve autonomy is applied to interplanetary communications systems.

INTRODUCTION

Autonomic communication is a critical capability in remote long-distance environments beyond the reach of real-time human interaction and manipulation. In Delay Tolerant Networks (DTNs) [1] found in deep-space, underwater, or at the North and South Poles, context-awareness enables autonomic optimization of operational efficiency and network lifetimes. Consumption of finite resources can be minimized by executing decision-making on-board and avoiding long round-trip latencies between mission-control on Earth. Autonomy allows the spacecraft to be self-guiding, with the effect of reducing operational costs, improving ability to respond to unexpected events, and maximizing opportunities for scientific discovery. Autonomy is therefore explored for application in mission designs and communication strategies in challenged long-distance networks. Swarms of networked components, for example, represent an autonomic scenario proposed for use in space [10] to enable scientific feats unachievable using traditional end-to-end networking techniques, including identification of asteroid size and

movement velocity. Swarm components communicate “on-the-fly” and dynamically adapt activity in response to data collected from other elements in the swarm. New protocols, which are designed to use context data and operate autonomically, have also been proposed to optimize communications operations in constrained networks. The Bundle protocol [8], for example, exploits store-and-forward transmission to overcome effects of intermittent connectivity, and the Licklider Transmission Protocol (LTP) [7] can be applied to transmit in blocks of compulsory and non-compulsory data to overcome the effects of high bit error rates. Taking into account both recent technology and protocol advancements, a research gap exists in that autonomy has not been applied to any great extent to *manage* interplanetary communications for both autonomic hardware operation and protocol use, allowing configuration of both to be managed remotely. The problem statement can therefore be summarized: dynamic and extreme conditions in deep-space require adaptability in the way node operation is defined, communication occurs, and protocols are configured. The cost of doing so can be optimized if performed autonomically to take into account local constraints and their impact on the mission, and transmission configuration should occur dynamically in response to real-time environmental conditions to optimize network utility. The research challenge therefore involves autonomic decisions regarding the most appropriate protocol features to use for each transmission, configuring it in the most suitable way, initiating transmission at the most appropriate times, and enabling dynamic node and communication stack reconfigurations in response to network operational environment change.

This investigates the deployment of an autonomic policy-based protocol stack middleware developed by the authors, the Context-Aware Broker (CAB) [5, 6], in a currently non-automated area of delay-tolerant networking. The middleware uses context-aware capabilities *on-the-fly* to enable autonomic transmission configuration such that application Quality of Service (QoS) may be achieved within environment constraints. For the purpose of our research, we automate and optimize choices within the transport layer due to protocol limitations in relation to long-distance

Author's Current Address:

C. Peoples, G. Parr, B. Scotney and A. Moore, School of Computing and Information Engineering, University of Ulster, Coleraine, Northern Ireland, UK.

Manuscript received March 15, 2010. Review handled by M. De Sanctis.

0885/8985/11/ \$26.00 © 2011 IEEE

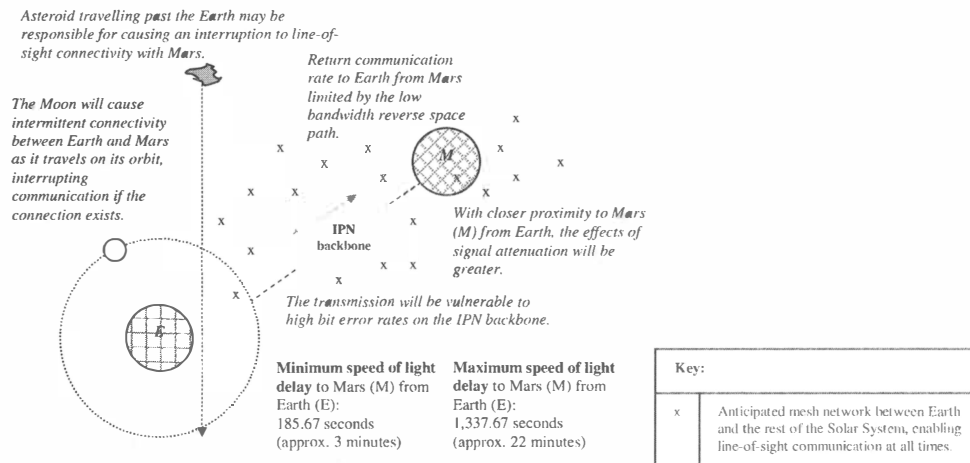


Fig. 1. Deep-Space Delay-Tolerant Network Scenario

networking. A range of transport protocols exist, however, while each offer a communication approach for a specific application or operation within a specific environment, a single protocol does not meet requirements for all applications and all environments. This implies that multiple protocols must be deployed on nodes when enabling self-configuring communication and an opportunity therefore exists to develop an intelligent selection and configuration function. Evidence supporting the need for autonomic communication configuration is expressed in [4]:

“Convergence Layer Adapters” reside on each Bundle Forwarder enabling decisions to be made on a selection of protocols, with routing and storage decisions achieved in relation to local applications and management processes. We therefore position the Context-Aware Broker as a Convergence Layer Adapter when related to Fall and Farrell’s (2008) model, with extensions which incorporate energy-efficiency and increased context capabilities.

The remainder is organized as follows: in the section entitled *State-of-the-Art Autonomic Management in Deep Space: A Case Study*, a state-of-the-art deep-space mission which could have benefited from increased autonomic capabilities is explored. This sets the scene for introduction of the autonomic development proposed by the authors in the *Context-Aware Broker for Interplanetary Communications Systems: The Design* section, where the research proposal and core components of its design are presented. In the *The Cost of Context* section, overhead costs associated with the integration of intelligence are evaluated theoretically in relation to the CAB’s design and the ways in which costs may be applied. Performance results are evaluated in the *Experiments and Results* section, where a simulated deployment of the Context-Aware Broker is tested in different scenarios. Finally, the *Conclusion & Future Work* section is presented.

STATE-OF-THE-ART AUTONOMIC MANAGEMENT IN DEEP-SPACE: A CASE STUDY

Context-awareness and autonomic operation allow reaction to network events which currently require recovery

using terrestrial-based intelligence. Several individual spacecraft deployed in deep-space have already experienced the benefits of autonomy and use intelligent techniques to, for example, locally discard images or datasets of lower quality as opposed to transmitting all collected over the long-distance link (e.g., a technique implemented during the National Aeronautics and Space Administration (NASA) Autonomous Sciencecraft Experiment [9]). Autonomy has also been used to direct a rover across planetary surfaces in response to terrain findings (e.g., which occurs during the NASA Opportunity traversal of Mars [1]). The NASA Spirit Rover, however, became trapped in sand on Mars in April 2009 and its autonomic capabilities were unable to cope with this exception condition. After several months of exploring release strategies remotely on Earth, at the time of this writing recovery of its mobility shows greater potential¹ but finite node power resources must also be considered to optimize the rover’s overall lifetime. The Martian winter begins in May 2010 and power levels which will invoke movement are already insufficient as of February 2010. Priorities during March 2010 therefore involve using the remaining energy to prepare Spirit for survival during the winter and recovery after winter. This includes uploading a communication strategy revised specifically for the winter period: spacecraft wake times are shortened, communication passes are deleted, and Spirit will exist in a hibernation mode, maintain a clock, and routinely check power levels until wake-up is possible. The Spirit mission is an example of a scenario where autonomic operation could have improved the speed of decision execution to optimize mission operation during an unpredicted event. Self-diagnosis and dynamic initiation of communication with mission-control to request assistance could have improved the power-efficiency of Spirit’s resources, allow quicker recovery initiation and optimization of resource consumption during restoration attempts.

Greater autonomic capability is increasingly important both for this and future missions: NASA’s Juno mission to

¹ More information on the NASA Spirit Mars Exploration Rover mission is available at: http://www.nasa.gov/mission_pages/mer/spirit-update.html.

Jupiter² will be approximately thirty minutes further from Earth than the Spirit Rover on Mars (Figure 1) when launched in 2011 and the communication path is propagated at the speed of light³. It will be exposed to longer round-trip communication times while waiting for commands from Earth during which finite network resources are consumed. It is in recognition of this fact that scientists pursue the development of autonomic communication and operation capabilities for interplanetary communications systems.

CONTEXT-AWARE BROKER FOR INTERPLANETARY COMMUNICATIONS SYSTEMS: THE DESIGN

The Context-Aware Broker (CAB) (Figure 2) is designed to achieve autonomic communications for node operation in deep-space. It is deployed in protocol stacks on all nodes in the network, collects data about the environment and from the application layer, and has the potential to influence decisions in the remaining layers of the stack using pre-defined policies. CAB execution initiates with *Data Collection* regarding the application and environment within which it operates. The CAB will collect a range of attributes relating to the environment (e.g., propagation delay between nodes), network (e.g., node battery levels and bandwidth availability), and applications (e.g., acceptable bit error rate and acceptable latency). *Data Validation* verifies as much as possible that collected data is accurate and that the decision-making process is working effectively. *Environment Validation*, for example, will check if any of the propagation distances between any nodes has been recorded as zero, refreshing collected data a limited number of times in the instance that it has to correct the value, before progressing on a best-effort basis when required. Within the *Evaluation* state, several phased processes review data to gain an appreciation of the performance achievable within operating constraints in relation to application requirements. This also comprises a further *Data Collection* phase where additional context data is *Inferred* (including e.g., application type of service and environment round-trip propagation delay) from that collected after progression through the *Phase 1* range of checks. During *Phase 2*, collected and inferred contextual data is evaluated to gain a greater understanding of current network conditions and the ability to meet application requirements within operating constraints. The relationship between round-trip propagation delay and maximum acceptable application latency, for example, is assessed

during *Phase 2 Evaluation* to determine if the minimum latency achievable between nodes will allow QoS to be achieved. *Phase 3 Evaluation* involves determining key requirements of a selected protocol, including, for example, the requirement for retransmissions and a store-and-forward reliability mechanism. Together, this set of procedures assists with the node state definition and *Protocol Configuration*. In response, node states may be manipulated and the transmission will be configured. Once sending begins, the CAB *Monitors* collected context data and *Predicts* future performance using attributes which include signal strength and propagation distance between mobile nodes. Network changes which force service levels to decline below an acceptable benchmark will be identified, with the opportunity to *Re-Configure* the communication configuration or node operation. The CAB's overall objective is to enable autonomic communication in DTNs, thus removing mission-specific configuration requirements. Individual communication requirements are transmission-specific, but optimizing reliability and sustainability in challenged environments are core CAB functionalities.

THE COST OF CONTEXT

Network transmissions are performed at an outlay for both network and user, with overheads measured in time and volume of data units. Default latency costs incurred during all transmissions, which include delays between application transmission requests and traffic receptions at destination, result from relationships between traffic volume, bandwidth availability, queuing and de-queuing serialization delay, end-to-end propagation delay, and data velocity⁴. Additional time costs may be incurred in selected scenarios when events, for example, loss of line-of-sight connectivity caused by orbital movement or node battery power failure, create an inability to communicate for known or unpredictable periods of time. Data costs are a function of traffic volumes, which can be consequential of relationships with bandwidth availability, protocol reliability, network bit error rate, and propagation distance⁵. Optimum transmissions will involve minimum data and latency costs when QoS is achieved.

While the Context-Aware Broker has been developed to restrict negative transmission costs, there is an inherent

$$Cost_{latency} = n - I (SenT_{net} / B) + D / D_{net} + n - I (RecvT_{net} / B) \quad (1)$$

Where: latency cost ($Cost_{latency}$) (Seconds) includes the delay to push a transmission ($SenT_{net}$) (bytes) from each node (n) (excluding the destination node) using the available bandwidth (B) (bytes per second) onto the link, propagation delay to propagate the link over the distance (D) (kilometers) between communicating nodes at the speed of light (D_{net}) (kilometers / second), and receive traffic ($RecvT_{net}$) (bytes) from each node excluding the source node) using available link bandwidth.

$$^5 \text{ When a reliable protocol is used: } Cost_{data} = SenT_{net} + A / [+R] \quad (2)$$

Where: data cost ($Cost_{data}$) (bytes) includes the initial transmission volume sent ($SenT_{net}$) (bytes) and the volume of traffic which acknowledges (A) (bytes) traffic received. There is an optional cost which may be incurred when a reliable protocol is used if data retransmissions (R) (bytes) occur. When an unreliable protocol is used:

$$Cost_{data} = SenT_{net} [-D_{drop}] \quad (3)$$

Where: data cost ($Cost_{data}$) (bytes) includes the initial transmission volume sent ($SenT_{net}$) (bytes), which will be reduced if network conditions cause data drop (D_{drop}) (bytes) to occur.

² More information on the NASA Juno mission to Jupiter is available at: http://www.nasa.gov/mission_pages/juno/main/index.html.

³ Mars and Jupiter are a maximum of approximately 401.3×10^6 and 968.1×10^6 kilometers from Earth respectively (according to: [National Aeronautics and Space Administration, "Mars Fact Sheet" (2004); available: <http://nssdc.gsfc.nasa.gov/planetary/factsheet/marsfact.html>] and [National Aeronautics and Space Administration, "Jupiter Fact Sheet" (2004); available: <http://nssdc.gsfc.nasa.gov/planetary/factsheet/jupiterfact.html>]). This relates to a latency of 31 minutes and 50 seconds when the link is traversed at the speed of light (assuming a speed of light propagation velocity of 299,792.46 kilometers/second and the absence of all operational and per hop node constraints).

minimum additional cost when it is applied, expressed as a function of the time to perform initial brokering processes. Maintaining a positive cost-benefit impact has therefore been an important design requirement of the CAB. The inclusion of all CAB costs is not a pre-defined process and depends on each scenario. CAB execution delay overhead is therefore the sum of one of the following cost patterns (with 1 representing CAB phase(s) invoked and 0 representing CAB phase(s) not invoked):

$$q0 = 1; q1 = 0; q2 = 0; q3 = 0 \quad (4)$$

$$q0 = 1; q1 = 1; q2 = 0; q3 = 0 \quad (5)$$

$$q0 = 1; q1 = 1; q2 = 1; q3 = 0 \quad (6)$$

$$q0 = 1; q1 = 1; q2 = 1; q3 = 1 \quad (7)$$

where:

$$q0 \text{ (seconds)} = (\text{DataCollection} + \text{Validation} + \text{Phase1Evaluation}) \quad (8)$$

$$q1 \text{ (seconds)} = (+ \text{InferringContextualData} + \text{Phase2Evaluation}) \quad (9)$$

$$q2 \text{ (seconds)} = (+ \text{Phase3Evaluation} + \text{Communication/NetworkConfiguration} + \text{Transmission} + \text{Monitoring}) \quad (10)$$

$$q3 \text{ (seconds)} = (+ \text{Alarms}) \quad (11)$$

Equation (12) reinforces the algorithm's incremental invocation and the pre-defined cost application order during contextual transmissions:

$$q0 >= q1 >= q2 >= q3 \quad (12)$$

Operational benefits of executing CAB *Alarms*, for example, cannot occur without *Data Collection*, *Validation*, and *Phase 1 Evaluation*. Performance benefits will therefore always occur at the minimum expense of overhead costs in pattern (4). If exposed to cost pattern (7), the transmission may achieve QoS: it will begin, but *Alarms*, for example, that propagation distance between mobile nodes is increasing, will indicate drops in performance. It is possible that transmission may be suspended until the negative influence ceases and is recovered from, or it can be *Re-Configured* – a protocol round-trip timeout, for example, can be adapted to accommodate the extended propagation distance. Contextual transmission expense may also be less than this cost, as shown in patterns (4), (5), and (6). If the transmission is exposed to cost pattern (6), QoS will be achieved as the transmission completes without *Alarms*. Bandwidth availability, node power, and signal strength, for example, will remain sufficient for transmission requirements throughout the duration of sending. If a transmission is exposed to cost pattern (5), however, QoS will not be achieved as execution is suspended after *Phase 2 Evaluation* and transmission will not begin. This may occur if, for example, the CAB identifies that battery power at one or more nodes on the network path is or will become insufficient as transmission progresses and the line-of-sight

connection will be lost during the communication. It is also possible that transmission costs include only *Data Collection*, *Validation*, and *Phase 1 Evaluation* (cost pattern (4)) because the CAB identifies during *Phase 1 Evaluation*, for example, that line-of-sight connectivity does not exist between communicating nodes, and if only these costs are incurred, QoS will not be achieved.

CAB cost overheads are incurred in relation to QoS-achievement probability to allow the cost-benefit relationship to be balanced. The cost of context is transmission-specific and, while costs are incurred in a pre-defined order during contextual transmissions, benefits vary in terms of improvement to throughput, sustainability, latency, power-efficiency, and/or accuracy. Additional costs without benefits occur when the CAB is not required to take action during pre-transmission phases, the environment remains stable as transmission progresses, and the configuration is the same as that selected when the CAB is not deployed. Although the algorithm design has been optimized to minimize cost impacts in such circumstances, the risk of this scenario occurring must be embraced to allow instances of autonomy leading to performance improvements.

EXPERIMENTS AND RESULTS

Simulation results from a CAB implementation reinforce the theoretical discussion of its cost-benefit impact in *The Cost of Context* section. There are three wireless nodes in the experimental scenario⁶ which represents two wireless nodes deployed on the surface of Mars communicating with a nearby landed manned spacecraft; node random mobility has been switched off and a multi-state error model affects communication links. The source node transmits 15,000 bytes of mission-critical File Transfer Protocol (FTP) application traffic, which cannot cope with any packet loss, and the initial protocol selected by the CAB is the Stream Control Transmission Protocol (SCTP) based on application requirements for transmission reliability and the vulnerable wireless environment. When the CAB is not applied to the scenario, there are nine SCTP packet de-queues before transmission ceases (Figure 3). Only three SCTP packets are en-queued at the destination node, and these are protocol control packets. SCTP timer expiration causes transmission to fail after 273 seconds due to the default maximum number of retransmissions being exceeded. In contrast when the CAB is deployed, the transmission has a higher level of success (Figure 4). Application transmission is suspended initially due to the high network bit error rate, and starts after 125 seconds. Once transmission begins, a CAB alarm after 350 seconds signals a disparity between de-queues and en-queues, acknowledging that protocol timers continue to expire over a period of time. This forces a handover from a reliable to unreliable protocol, and application transmission

⁶Nodes are positioned at X, Y, Z grid coordinates (with grid propagation distance measured in meters): node 0: 1, 2, 0; node 1: 16, 13, 0; node 2: 25, 39, 0.

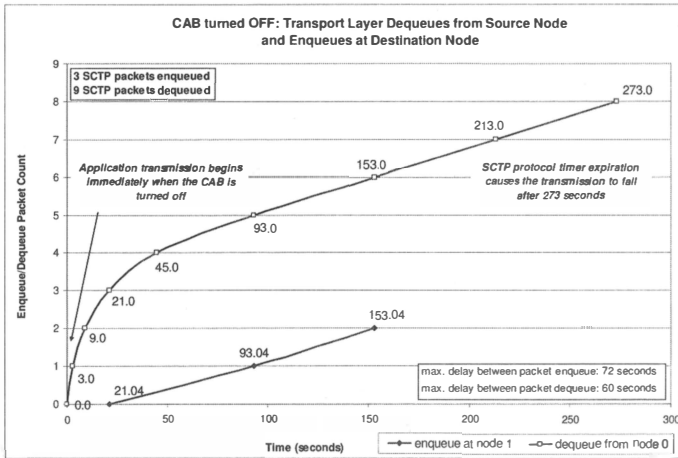


Fig. 3. Results 1A: De-queues and En-queues at Transport Layer when CAB turned Off

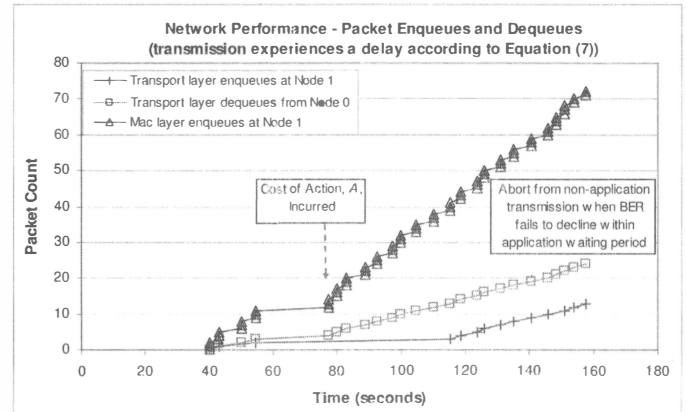


Fig. 5. Results 2A: Cost-Benefit Impact Scenario: Non Mission-Critical Transmission

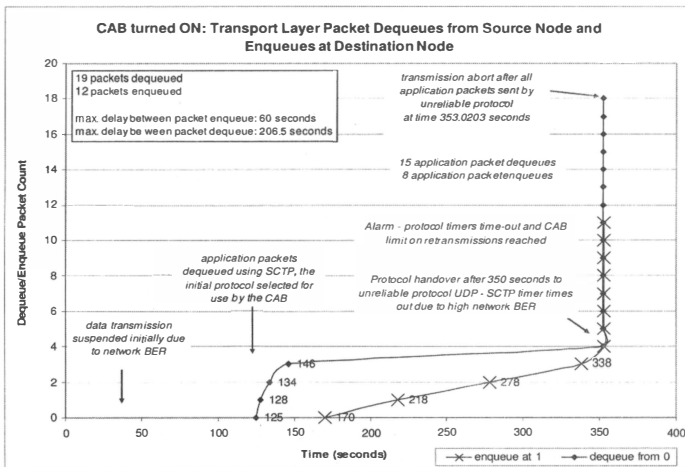


Fig. 4. Results 1B: De-queues and En-queues at Transport Layer when CAB turned On

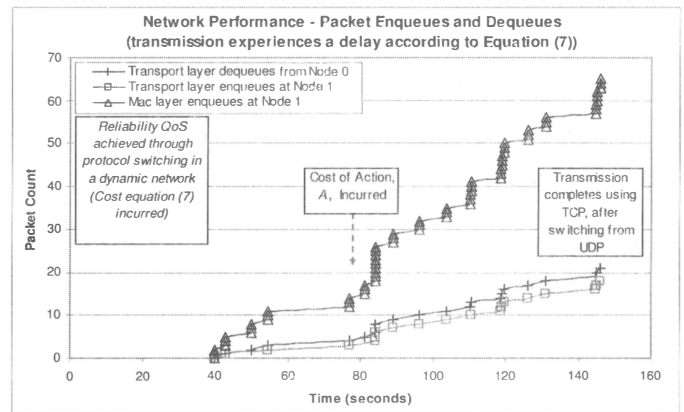


Fig. 6. Results 2B: Cost-Benefit Impact Scenario: Mission-Critical Transmission

re-starts at this time. Fifteen packets of application traffic are de-queued and eight packets are en-queued. While full application transmission reliability requirements are not achieved, the CAB allows sending to continue on a best-effort basis due to application mission-criticality and an inability to guarantee that the network condition will improve within the application's acceptable transmission latency.

Another scenario is used to represent CAB performance where cost overheads are responsible for sustaining the longer-term condition of the network. Decisions represented in these experimental results are based on three application attributes, including application mission-criticality (true or false), maximum acceptable bit error rate (BER_{acc}) (percentage of packets lost per second), and maximum acceptable latency (seconds).

All other context attribute values remain constant and do not drive network reaction across the experiments detailed herein. Transmission decisions in this scenario are based on relationships between network bit error rate (BER) and

acceptable application bit error rate, and transmission duration in relation to acceptable application latency.

As in the previous scenario, there are two wireless nodes deployed on the surface of Mars communicating with a nearby landed manned spacecraft⁶, and a file transfer volume of 15,000 bytes is sent between end-points in packets of 1,000 bytes. The application has a required transmission latency of 60 seconds and is transmitted in intervals of 0.2 seconds. Dynamic events during the transmission include variation in bit error rate⁷ (error model 1 or 2) and node mobility⁸ (on or off). Network performance is measured using the difference between en-queues at the destination node and de-queues from the source node, both at the

⁶Nodes are positioned at X, Y, Z grid co-ordinates (with grid propagation distance measured in meters): node 0: 1, 2, 0; node 1: 16, 13, 0; node 2: 25, 39, 0.

⁷Error model 1: Multi-state error rates (percentage of packets lost per second): rate 1: .1 pkt; rate 2: .5 pkt; rate 3: .3 pkt; rate 4: 0 pkt; error rate periods (proportion of overall transmission duration) rate 1: .2; rate 2: .2; rate 3: .3; rate 4: .3. Error model 2: Uniform error rate (percentage of packets lost per second): 4 pkt.

⁸Node mobility path (specified in re-located X, Y, Z grid coordinates): after 20 seconds: node 2: 500, 570, 4; after 40 seconds: node 1: 700, 738, 10; after 80 seconds: node 1: 35, 45, 10.

transport layer, and the number of en-queues at the MAC layer of the destination node. Performance of the CAB is also measured according to its brokering execution latency. Three scenarios test performance in terms of the CAB's overhead and associated benefits: in transmission 1, the application is not mission-critical and cannot cope with data loss (e.g., a data stream reporting soil mineral properties for scientific discoveries); in transmission 2, the application is mission-critical and cannot cope with data loss (e.g., astronaut biometric data from an attached sensor to spacecraft control as he traverses the Martian plains); and in transmission 3, the application is not mission-critical and can cope with up to 0.1 packet loss per second (e.g., an image of the Martian terrain). Each scenario is tested with the different error and mobility conditions⁹. In all scenarios, a transport protocol without reliability mechanisms is selected initially, given environmental conditions at the time of making the decision. CAB reactions during transmission include changing between protocols based on their reliability and suspending sending to observe if negative network conditions change within a delay which is acceptable for the application.

Performance plots chart en-queue and de-queue counts at and from the transport layer of source and destination nodes, and en-queues at the MAC layer of the destination node when the CAB is deployed. Cost pattern (7) is incurred in Figures 5 and 6 because $BER > BER_{acc}$, although reliability QoS is achieved only in Figure 6. Figure 5 shows network performance when nodes are mobile and a multi-state error model is used. The transmission begins after 40 seconds and a non-application stream starts after 80 seconds once the CAB identifies a BER greater than acceptable for the application and subsequently suspends its transmission. After 160 seconds, the maximum application waiting period has been exceeded and the CAB's consideration for transmission re-start is suspended. This is calculated on the basis that the application waiting period is 60 seconds and therefore expires after 140 seconds, but is not flagged to the CAB until its next scheduled check after 160 seconds. The CAB can suspend transmission as the application is not mission-critical, attempting to maximise both latency and reliability QoS simultaneously when it detects a BER which is higher than acceptable. It also attempts to maximize latency QoS by transmitting using an unreliable protocol and waits for a period until the BER declines so that it can continue transmitting. However, it does not decline to a satisfactory level and the CAB therefore achieves its objectives of prioritizing resource consumption conservation when neither the required latency nor reliability QoS will be achieved within environment constraints.

Figure 6 shows performance when nodes are not mobile and the multi-state error model is used. In this scenario, the transport protocol handover from UDP to TCP occurs after 80 seconds because the BER is unacceptable for the

application and it is mission-critical. In this scenario, reliability QoS is achieved, but latency QoS is compromised. Reliability QoS is prioritized, however, given the mission-criticality of the application.

The results can also be considered in terms of CAB delay¹⁰, which includes the time for the algorithm to execute, simulation configuration to occur and transmission to complete when taking into account monitoring, prediction, or intermediary actions. Maximum execution delay occurs when the application is mission-critical, cannot cope with data drop, and the uniform error model is used. Soon after beginning transmission, protocol handover to TCP occurs because of the detected change in the error rate once transmission begins. The reliable transport protocol therefore attempts to ensure accurate and reliable transmission in the challenged environment for the mission-critical application and retransmissions contribute to increased execution time.

From the results, it is possible to determine the number of scenarios in which QoS is achieved and the CAB's cost-benefit impact is balanced.

Where fifteen packets of application traffic are en-queued at the destination node transport layer, as in Figure 6, reliability QoS is achieved. Latency QoS, however, is not achieved as the CAB overrides this performance attribute on the basis that the application is mission-critical, with the objective of ensuring transmission reliability. The remainder of scenarios which are not mission-critical do not achieve reliability QoS, as in Figure 5, as the CAB restricts transmission to ensure that they meet latency requirements. When it detects that the BER is unacceptable for the application, it halts transmission and monitors the network conditions. When it detects that the bit error rate is not declining to a suitable level within an acceptable time period, it suspends consideration of transmission re-start until another timeout occurs. This decision has been taken to minimise cost overheads and improve resource availability for mission-critical applications.

In general, to achieve the benefits of deploying context, greater cost overhead can be expected. The decision to consume network resources by initiating a transmission, for example, is evaluated using the application transmission priority and probability of achieving QoS within environment constraints so that the cost-benefit relationship may be balanced. In the results shown, cost pattern (7) is incurred in Figure 6, and the benefit of the contextual process is reflected in the goodput performance improvement. Scenarios are also observed where costs are incurred without direct transmission benefits, and are a consequence of the CAB prioritizing resource consumption conservation in relation to non-mission-critical application transmission. Such scenarios are allowed to occur to optimize overall lifetimes of the network so that resources are available for critical

⁹ Each with: (a) node mobility on, error model 1; (b) node mobility on, error model 2; and (c) node mobility off, error model 1.

¹⁰ Average (of eight) CAB execution latencies (microseconds):
Transmission 1: a) 4,073,263; b) 3,974,633; c) 3,736,504.
Transmission 2: a) 3,973,292; b) 4,850,320; c) 3,148,330.
Transmission 3: a) 3,935,670; b) 4,089,007; c) 4,344,603.

transmissions when required, and a longer-term cost-benefit balance is achieved.

CONCLUSION & FURTHER WORK

The Context-Aware Broker's aim is to enable transmissions to maximize performance and minimize costs by empowering it with the ability to cope with unpredictable environment events and adapt transmission configuration at the node. In addition, the CAB minimizes wasteful consumption of resources by suspending sending when it cannot achieve application QoS at that point in time to improve resource availability for mission-critical transmissions when required in the future. Results from a deployment of the CAB confirm that it introduces improved ability to achieve application QoS and/or resource consumption conservation. These achievements occur as a consequence of the Context-Aware Broker prioritizing aspects of QoS: to achieve reliability QoS, for example, latency QoS is compromised. While both aspects of QoS are ideally achieved, the CAB prioritizes between a competing set of requirements to optimize performance within the resource-constrained environment.

The next stage of this work involves investigating the point at which Context-Aware Broker overhead leads to efficiency loss. These future investigations may indicate a network size (measured in either number of nodes or propagation distance) at which use of the CAB leads to performance losses and should no longer be considered for deployment.

REFERENCES

[1] M. Bajracharya, M.W. Maimone and D. Helmick,
Autonomy for Mars Rovers: Past, Present, and Future,
IEEE Computer Magazine, 2008, pp. 44-50.

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott,
K. Fall and H. Weiss,

Delay-Tolerant Networking Architecture,
Request for Comments 4838, 2007.

[3] S. Chien, R. Knight, A. Stechert, R. Sherwood and G. Rabideau,
Integrated Planning and Execution for Autonomous Spacecraft,
IEEE Aerospace and Electronic Systems Magazine,
2009, pp. 23-30.

[4] K. Fall and S. Farrell,
DTN: An Architectural Retrospective,
IEEE Journal on Selected Areas in Communications,
Vol. 26, Issue 5, 2008, pp. 828-836.

[5] C. Peoples, G. Parr, B. Scotney and A. Moore,
A Reconfigurable Context-Aware Protocol Stack for
Interplanetary Communication,
in *Proc. 1st IEEE Int. Conf. Satellite and Space
Communications*, 2007, pp. 281-285.

[6] C. Peoples, G. Parr, B. Scotney and A. Moore,
A Context-Aware Policy-Based Framework for Self-Management
in Delay-Tolerant Networks (A Case Study for Deep Space
Exploration),
accepted for publication in *IEEE Communications Magazine
Network and Service Management Series*,
published July 2010.

[7] M. Ramadas, S. Burleigh and S. Farrell,
Licklider Transmission Protocol – Specification,
Request for Comments 5326, 2008.

[8] K. Scott and S. Burleigh,
Bundle Protocol Specification,
Request for Comments 5050, 2007.

[9] R. Sherwood, S. Chien, D. Tran, B. Cichy, R. Castano, A. Davies
and G. Rabideau,
Preliminary Results of the Autonomous Spacecraft Experiment,
IEEE Aerospace Conference, 2004.

[10] W.F. Trzaskowski, M.G. Hinchey, J.L. Rash and C.A. Rouff,
Autonomous and Autonomic Systems: A Paradigm for Future
Space Exploration Missions,
IEEE Transactions on Systems, Man, and Cybernetics,
Vol. 36, 2006, pp. 279-291.